YAHOO! search

Search Home - Yahoo! -

Your Search: "object oriented" "record oriented" database Search

Advanced Web Search
Preferences

Web | Directory | News NEW! | Yellow Pages | Images

powered by hp

TOP 20 WEB RESULTS out of about 937

1. Concurrency and Transaction Management in an **Object Oriented ...** (PDF)
   ... Is the coupling of **Object Oriented** (OOP) Programming principles with **Database** Management System ... 2,6 Relational DB Uses **record-oriented** model Data ...
   students.depaul.edu/~jtorres4/se690/Presentation1.pdf view as html | more results from this site

2. Citations: Semantic Concurrency Control in **Object--Oriented ...**
   ... Semantic Concurrency Control in **Object-Oriented Database** Systems", in IEEE Data Engineering 1993, Vienna, Austria. Admissible **Record-Oriented** Evaluation Plans ...
   citeseer.nj.nec.com/context/101846/153985 cached | more results from this site

3. Data Conversion from **Object-Oriented** to Relational **database** and ... (PDF)
   ... aspect of interest to the **database** application ... In the classical **record-oriented** data models (Relational, Network ... In an **Object Oriented** system, real-world objects ...
   www.cse.cuhk.edu.hk/~acm-hk/activity/pg/cityu-ptpang.pdf view as html

4. ADBIS 1997: 133-140
   ... are used to access the **database** during the ... cases more efficiently, in a **record-oriented** way. ... Query Optimization for Methods in **Object-Oriented Database** Systems ...
   www.informatik.uni-trier.de/~ley/db/conf/adbis/FischerA97.html cached | more results from this site

5. Introduction to **object-oriented database** management systems (...
   ... **database** management systems (OODBMS) in terms of what is the value added beyond **record-oriented database** systems and **object-oriented** programming languages. ...
   portal.acm.org/citation.cfm?id=157770&jmp=indexterms&dl=GUIDE&dl=ACM&CFID=11111111&CFTOKEN=2222222
   more results from this site

6. Geography 685 – Intermediate GIS Why Objects? **Object-Oriented ...** (PDF)
   ... of the real world • **Record-oriented** – close to the ... direct correspondence **Object-Oriented** Approach • Alternative ... Databases and **database** management systems ...
   www.geography.ohio-state.edu/classdocs/Murray/685pdf/web2b.PDF view as html

7. Jeff Johnson
   ... author points out, in the past most databases were **record-oriented** in nature, rather than **object-oriented** or semantic in nature. A record-base **database** is one ...
   www.cs.byuh.edu/research/johnson/serialization/papers/ooperspective.html cached

8. The Past 20 Years of **Database** Research David Maier Oregon ... (PDF)
   ... Past 20 Years of **Database** Research David ... File Systems Device Independence **Record- Oriented** DBMSs Physical ... Systems Deductive and **Object- Oriented** DBMSs Logical ...
   www.cse.ogi.edu/2020vision/Maier.pdf view as html | more results from this site

9. **Database** Systems
   ... to the best **record-oriented database** systems ... By 1990, virtually all **database** systems provided an ... network, hierarchical and **object-oriented database** systems, in ...
   www.cs.washington.edu/homes/lazowska/cra/database.html cached

10. **Object-Oriented Database** Systems (OODBS) Overview (PDF)
    ... **database** UoD **record oriented** data model 1:N **object oriented** data model 1:1 **database database** 2002 Pål Halvorsen INF 212 – **database** theory Advantages of OO ...
    www.ifi.uio.no/inf212/handouts/uke09-1.pdf view as html

11. **Object-Oriented Database** Systems: A Survey (PDF)
    ... They all focused on **record-oriented** information and ... In addition, new **database** applications need more ... Finally, **Object-Oriented** data model popularized starting ...
    www.cse.ucsc.edu/~lcx/courses/cmps277/cmps277-project.pdf view as html

12. PKS Software, iSeries (AS/400) tools for migration, reengineering ...
    ... driver for MS SQL interface · prototype of **record oriented** access patterns to DB2/UDB **database**. ... feel was followed by a more **object oriented** front-end ...
    www.pks.de/corporation/experts.htm cached

13. **Database** Management Systems - Prof. Holowczak - Week 14
    ... **Record-oriented** data models are effective when applied ... However, for advanced **database** applications such as ... **Object Oriented** Data Models and OODBMS are designed ...
    cisnet.baruch.cuny.edu/holowczak/classes/4300/week14.html cached

14. Architettura di Sistemi **Record-Oriented**, Parte 1
    ... In termini di Systematic **Object Oriented** Design [Pes99 ... più stabili nelle applicazioni **record-oriented** è l ... interessante è che i **database** relazionali, nonché ...
    www.eptacom.net/pubblicazioni/pub_it/recor1.html cached | more results from this site

15. BCS: eWiC Archive 1997 - Advances in Databases
    ... View Systems Admissible **Record-Oriented** Evaluation Plans ... A Validation System for **Object Oriented** Specifications of ... Paradigm The Sybil **Database** Integration and ...
    www1.bcs.org.uk/portal/homepage.asp?sectionid=715 cached | more results from this site

16. Lecture 14
... **Object-oriented** Approach. an attempt to improve modeling of the real world; **record-oriented** -> close to the computer; ... Motivation for **object-oriented database**. ...
whopper.sbs.ohio-state.edu/classpages/courses/g685/week8.html cached

17. **Object Oriented** Analysis and Design (MICROSOFT POWERPOINT)
... Objects that require persistent storage, such as a **database**. ... is required for resolving mismatches occurring between **record oriented** and **object-oriented** data ...
www-ec.njit.edu/~gblank/cis683/Larman%20Chapter%2034.ppt view as html

18. **Object Oriented** Databases (PDF)
... because a large amount of data is not '**record oriented**'. **Object Oriented** Databases http://www.dis.port.ac.uk/~chandler/OOLectures/**database/database**.htm (4 of 14 ...
www.jeffmcneill.com/bus311/oodbs.pdf view as html

19. **DATABASE** SYSTEMS (MICROSOFT POWERPOINT)
9. **Object-Oriented** DB and Multimedia DB. 9 -. New **Database** Applications. Properties of Conventional Databases. - Fixed size **record oriented**. - Small data item. ...
comedu.kongju.ac.kr/csk/database/9db_note.ppt view as html

20. Information Retrieval: Beyond Z39.50 (MICROSOFT POWERPOINT)
... discover real data-base schema. Real **database** query. ... Abstract. **record oriented**, hierarchical, relational, **object oriented** data models. Dynamic Infrastructure. ...
www.sbg.ac.at/geo/eogeo/authors/ward/wardfinnigan/wardfinnigan.ppt view as html

Results Page:
1 2 3 4 5 6 7 8 9 10 ▶ **Next**

Web **Directory** **News** NEW! **Yellow Pages** **Images**

Your Search: "object oriented" "record oriented" database Search Advanced Web Search Preferences

Search Technology provided by Google

Database Systems:
A Textbook Case of Research Paying Off

James N. Gray
Senior Researcher
Microsoft Corporation

Industry Profile

The database industry generated about $7 billion in revenue in 1994 and is growing at 35% per year. Among software industries, it is second only to operating system software. All of the leading corporations in this industry are US-based: IBM, Oracle, Sybase, Informix, Computer Associates, and Microsoft. In addition, there are two large specialty vendors, both also US-based: Tandem, selling over $1 billion per year of fault-tolerant transaction processing systems, and AT&T-Teradata, selling about $500 million per year of data mining systems.

In addition to these well-established companies, there is a vibrant group of small companies specializing in application-specific databases -- text retrieval, spatial and geographical data, scientific data, image data, and so on. An emerging group of companies offer object-oriented databases. Desktop databases are another important market focused on extreme ease-of-use, small size, and disconnected operation.

A relatively modest federal research investment, complemented by an also-modest industrial research investment, has led directly to our nation's dominance of this key industry.

Historical Perspective

Companies began automating their back-office bookkeeping in the 1960s. COBOL and its record-oriented file model were the work-horses of this effort. Typically, a batch of transactions was applied to the old-tape-master, producing a new-tape-master and printout for the next business day.

During this era, there was considerable experimentation with systems to manage an on-line database that could capture transactions as they happened, rather than in daily batches. At first these systems were ad hoc, but late in the decade "network" and "hierarchical" database products emerged. A network data model standard (DBTG) was defined, which formed the basis for most commercial systems during the 1970s. Indeed, in 1980 DBTG-based Cullinet was the leading software company.

However, there were some problems with DBTG. DBTG used a low-level, record-at-a-time procedural language. The programmer had to navigate through the database, following pointers from record to record. If the database was redesigned, as they often are over a decade, then all the old programs had to be rewritten.

The "relational" data model, enunciated by Ted Codd in a landmark 1970 article, was a major advance over DBTG. The relational model unified data and metadata so that there was only one form of data representation. It defined a non-procedural data access language based on algebra or logic. It was easier for end-users to visualize and understand than the pointers-and-records-based DBTG model. Programs could be written in terms of the "abstract model" of the data, rather than the actual database design; thus, programs were insensitive to changes in the database design.

The research community (both industry and university) embraced the relational data model and extended it during the 1970s. Most significantly, researchers showed that a high-level relational database query language could give performance comparable to the best record-oriented database systems. This research produced a generation of systems and people that formed the basis for IBM's DB2, Ingres, Sybase, Oracle, Informix and others. The SQL relational database language was standardized between 1982 and 1986. By 1990, virtually all database systems provided an SQL interface (including network, hierarchical and object-oriented database systems, in addition to relational systems).

Meanwhile the database research agenda moved on to geographically distributed databases and to parallel data access. Theoretical work on distributed databases led to prototypes which in turn led to products. Today, all the major database systems offer the ability to distribute and replicate data among nodes of a computer network.

Research of the 1980s also showed how to execute each of the relational data operators in parallel -- giving hundred-fold and thousand-fold speedups. The results of this research are now beginning to appear in the products of several major database companies.

Three Case Studies

The government has funded a number of database research efforts from 1970 to the present. Projects at UCLA gave rise to Teradata and produced many excellent students. Projects at CCA (SDD-1, Daplex, Multibase, and HiPAC) pioneered distributed database technology and object-oriented database technology. Projects at Stanford created deductive database technology, data integration technology, and query optimization technology. Work at CMU gave rise to general transaction models and ultimately to the Transarc Corporation. There have been many other successes from AT&T, the University of Texas at Austin, Brown, Harvard, Maryland, Michigan, MIT, Princeton, and Toronto, among others. It is not possible to enumerate all the contributions here, but we shall highlight three representative research projects that had major impact on the industry.

Ingres

Project Ingres started at UC Berkeley in 1972. Inspired by Codd's work on the relational database model, several faculty members (Stonebraker, Rowe, Wong, and others) started a project to design and build a relational database system. Incidental to this work, they invented a query language (QUEL), relational optimization techniques, a language binding technique, and interesting storage strategies. They also pioneered work on distributed databases.

The Ingres academic system formed the basis for the Ingres product now owned by Computer

Associates. Students trained on Ingres went on to start or staff all the major database companies (AT&T, Britton Lee, HP, Informix, IBM, Oracle, Tandem, Sybase). The Ingres project went on to investigate distributed databases, database inference, active databases, and extensible databases. It was rechristened Postgres, which is now the basis of the digital library and scientific database efforts within the University of California system. Recently, Postgres spun off to become the basis for a new object-relational system from the startup Illustra Information Technologies.

System R

Codd's ideas were inspired by seeing the problems IBM and its customers were having with the DBTG network data model and with IBM's product based on this model (IMS). Codd's relational model was at first very controversial; people thought that the model was too simplistic and that it could never give good performance. IBM Research management took a gamble and chartered a 10-person effort to prototype a relational system based on Codd's ideas. This group produced a prototype, System R, that eventually grew into the DB2 product series. Along the way, the IBM team pioneered ideas in query optimization, data independence (views), transactions (logging and locking), and security (the grant-revoke model). In addition, the SQL query language from System R was the basis for the standard that emerged.

The System R group went on to investigate distributed databases (project R*) and object-oriented extensible databases (project Starburst). These research projects have pioneered new ideas and algorithms. The results appear in IBM's database products and in those of other vendors.

Gamma

During the 1970s there was great enthusiasm for database machines -- special-purpose computers that would be much faster than general-purpose systems running conventional databases. The problem was that general purpose systems were improving at 50% per year, so it was difficult for customized systems to compete with them. By 1980, most researchers recognized the futility of special-purpose approaches, and the database machine community switched to research on using arrays of general purpose processors and disks to process data in parallel. The University of Wisconsin was home to the major proponents of this idea in the US. Funded by the government and industry, they built a parallel database machine called Gamma. That system produced ideas and a generation of students who went on to staff all the database vendors. Today the parallel systems from IBM, Tandem, Oracle, Informix, Sybase, and AT&T all have a direct lineage from the Wisconsin research on parallel database systems. The use of parallel database systems for data mining is the fastest-growing component of the database server industry.

The Gamma project evolved into the Exodus project at Wisconsin (focusing on an extensible object oriented database). Exodus has now evolved to the Paradise system which combines object-oriented and parallel database techniques to represent, store, and quickly process huge earth-observing satellite databases.

The Future

Database systems continue to be a key aspect of Computer Science & Engineering today. Representing knowledge within a computer is one of the central challenges of the field. Database research has focused primarily on this fundamental issue. Many universities have faculty investigating these problems and offer courses that teach the concepts developed by this research program.

There continues to be active and valuable research on representing and indexing data, adding inference to data search, compiling queries more efficiently, executing queries in parallel, integrating data from heterogeneous data sources, analyzing performance, and extending the transaction model to handle long transactions and workflow (transactions that involve human as well as computer steps). The availability of very-large-scale (tertiary) storage devices has prompted the study of models for queries on very slow devices.

In addition, there is great interest in unifying object-oriented concepts with the relational model. New datatypes (image, document, drawing) are best viewed as the methods that implement them rather than the bytes that represent them. By adding procedures to the database system, one gets active databases, data inference, and data encapsulation. This object-oriented approach is an area of active research and ferment both in academe and in industry.

A very modest research investment produced American market dominance in a $7 billion industry -- creating the ideas for the current generation of products, and training the people who built those products. Continuing research is creating the ideas and training the people for the next product generation.

The Author

Dr. James N. Gray is one of the world's leading experts on database and transaction processing computer systems. Over the past three decades he has worked for IBM, Tandem, and Digital Equipment Corporation on systems including System R, SQL/DS, DB2, IMS-Fast Path, Encompass, NonStopSQL, Pathway, TMF, Rdb, DBI, and ACMS -- systems that have defined the progress of the field.

Dr. Gray holds doctorates from U.C. Berkeley and the University of Stuttgart. He is a Fellow of the ACM, a member of the National Research Council's Computer Science and Telecommunications Board, Editor-in-Chief of the *VLDB Journal*, Editor of the Morgan Kaufmann series on Data Management, editor of the *Performance Handbook for Database and Transaction Processing Systems*, co-author of *Transaction Processing Concepts and Techniques*, and serves on Objectivity's Technical Advisory Board.

In 1995, following six months as McKay Fellow in U.C. Berkeley's Computer Science Division, he joined Microsoft to establish a San Francisco Bay Area laboratory focusing on making Microsoft data servers more scaleable, manageable, and fault tolerant.

Back to Computing Research: Driving Information Technology and the Information Industry Forward (http://cra.org/research.impact)

Contributions to this document by Philip A. Bernstein, David DeWitt, Michael Stonebraker, and Jeffrey D. Ullman are appreciated.